



Dieses Dokument wird unter folgender [Creative-Commons](http://creativecommons.org/licenses/by-nc-nd/2.0/de/)-Lizenz veröffentlicht: <http://creativecommons.org/licenses/by-nc-nd/2.0/de/>

Programmieren lernen in kollaborativen Lernumgebungen

Ralph Kölle¹, Glenn Langemeier¹, Wolfgang Semar²

¹ Universität Hildesheim
Fachbereich für Informations- und
Kommunikationswissenschaften
31141 Hildesheim
{ralph,glenn}@vitaminl.de

² Universität Konstanz
Fachbereich Informatik und
Informationswissenschaft
78457 Konstanz
wolfgang.semar@uni-konstanz.de

Zusammenfassung

Im Sommersemester 2005 fand (erstmalig) eine Kooperationsveranstaltung „Einführung in die objekt-orientierte Programmiersprache Java“ der Universitäten Konstanz und Hildesheim statt. Traditionelle Lehrveranstaltungen zum Thema Programmierung zeichnen sich neben der Wissenvermittlung (i.d.R. durch Vorlesung) durch einen hohen Grad an Praxisanteilen aus. Dazu arbeiten Teams in Tutorien gemeinsam an Übungsaufgaben. Der Einsatz der Systeme „K3“ (kollaboratives Wissensmanagement in Lernumgebungen, Konstanz) und „VitaminL“ (synchrone, kurzzeitige Bearbeitung von Programmieraufgaben, Hildesheim) ermöglicht nun die Übertragung einer solchen Veranstaltung ins Virtuelle. Lerngruppen arbeiten standort-übergreifend sowohl asynchron als auch synchron zusammen. Dieser Beitrag liefert neben dem Erfahrungsbericht der Kooperationsveranstaltung im ersten Teil einen Einblick in die Konzeption, Implementierung und Evaluation des VitaminL-Systems. Im zweiten Teil wird die Entwicklung eines Kennzahlensystems zur Leistungsevaluation kollaborativer Gruppenarbeit aufgezeigt.

1 Entwicklung und Evaluation einer kollaborativen Lernumgebung

1.1 Einführung

Die computer-basierte Unterstützung von Arbeitsgruppen hat mittlerweile einen hohen Standard erreicht, etablierte Groupware-Systeme wie Lotus Notes, Microsoft Exchange sowie das Open-Source-Projekt Kolab zeugen davon. Auch das Lernen in der Gruppe hat sich gegenüber dem individuellen Lernen als überlegen erwiesen (vgl. Schulmeister 2001, S. 196), jedoch bieten verfügbare Systeme wie bspw. Lotus Virtual Classroom, Blackboard oder Hyperwave eLearning Suite – trotz zweifelsohne hohen Funktionsumfangs – kaum mehr als eine technische Plattform, in der Lernmaterialien bereitgestellt werden und Lehrer und Lerner sich treffen und austauschen können. Der eigentliche Lernprozess wird in der Regel nur insofern unterstützt, indem Werkzeuge zu dessen Strukturierung und zur Überprüfung des Lernerfolgs angeboten werden.

Für die adäquate Unterstützung von (virtuellen) Lerngruppen in Problemsituationen werden spezielle CSCL-Systeme, die sog. intelligenten tutoriellen Systeme (ITS), benötigt. Durch Beobachtung der Teilnehmer während der Zusammenarbeit ziehen Systeme wie bspw. COMET, Algebra-JAM oder C-CHENE Rückschlüsse auf aktuelle Problemsituationen oder auf die Zusammensetzung des Teams (auf der Grundlage eines geeigneten Rollenmodells) und ermitteln aus den Eingangsdaten geeignete Strategien für gezielten tutoriellen Support.

Mit dem Projekt VitaminL¹ (Virtuelle Teams: Analyse und Modellierung in netzbasierten Lernumgebungen) wird derzeit ein CSCL-System entwickelt, das virtuelle Teams bei der kurzzeitigen, synchronen Zusammenarbeit im Bereich der objektorientierten Programmierung in Java unterstützen soll. Um einen angemessenen Support bieten zu können, wird die personelle Zusammensetzung berücksichtigt: zuviel Hilfe mindert den Lernerfolg, zuwenig oder falsche Hilfe reduziert die Motivation.

¹ Mehr zu dem Projekt unter <http://www.vitaminl.de>

1.2 VitaminL – Entwicklung einer IDE

Die Grundlage des CSCL-Systems von VitaminL bildet ein technisches System zur verteilten, synchronen Programmierung in Java, die *VitaminL-IDE*. Die Entstehung dieser Applikation ist durchgängig von Evaluationen begleitet und soll nachfolgend anhand wichtiger Eckpunkte skizziert werden.

1.2.1 Ausgangssituation

An der Universität Hildesheim finden in jedem Semester Veranstaltungen zur Einführung in die Programmierung mit der objektorientierten Programmiersprache Java statt. Pro Semester nehmen – je nach Studiengang – ca. 50 bis 80 Studierende an der jeweiligen Veranstaltung teil. Die Teilnehmer befinden sich im Grundstudium und haben in der Regel keine Vorkenntnisse oder nur geringe Programmiererfahrungen.

Der Lerninhalt wird in Form einer Vorlesung zunächst theoretisch vermittelt und anschließend in Tutorien praktisch angewendet. Kleingruppen von drei bis vier Studierenden erhalten mit geeigneten Übungsaufgaben Gelegenheit, kleine Probleme in Java innerhalb der Gruppe zu bearbeiten und dabei das notwendige Wissen aufzubauen. Der Bearbeitungsaufwand pro Aufgabe liegt anfangs bei ca. ein bis zwei Stunden.

1.2.2 Probleme

In dem oben geschilderten Szenario treten immer wieder Probleme auf, die oft nur unter Zuhilfenahme eines (realen) Tutors geklärt werden können. Diese lassen sich unterteilen in Probleme technischer, inhaltlicher und organisatorischer Natur.

1.2.2.1 Technische Probleme

Die ersten Hürden, die Anfänger einer Programmiersprache zu meistern haben, bestehen darin, das zugehörige Programmiersystem – bestehend aus Compiler, Interpreter, Debugger etc. – auf dem eigenen Arbeitsrechner zu installieren. Auch die anschließende Bedienung stellt gerade Anfänger vor Probleme, da die ersten Java-Programme in einer Shell übersetzt und ausgeführt werden; in Zeiten von Betriebssystemen mit ausgefeilter graphischer Benutzungsoberfläche ist diese Vorgehensweise zunächst mehr als gewöhnungsbedürftig.

1.2.2.2 Inhaltliche Probleme

Während der Programmierung treten üblicherweise weitere Probleme auf, die sich begründen lassen durch die Programmiersprache selber: da Java – wie alle anderen Programmiersprachen auch – eine formale Sprache ist, die eine

wohldefinierte Syntax und Semantik besitzt, ist der Umgang damit anfangs recht ungewohnt. Auch fehlt den Studierenden üblicherweise die Erfahrung beim Formulieren von Ansätzen für Problemlösungen und von Algorithmen. Die resultierenden Fehlermeldungen des Compilers sind oft nur schwer verständlich, Laufzeitfehler stellen die angehenden Programmierer vor weitere Probleme.

1.2.2.3 Organisatorische Probleme

Die Gruppenarbeit wirft weitere Probleme auf: zunächst müssen sich alle Mitglieder einer Gruppe auf einen gemeinsamen Termin einigen – üblicherweise finden solche Treffen in einem der in der Universität verfügbaren Rechnerräume statt. Während der gemeinsamen Arbeit an den Quelltexten müssen sich die Mitglieder auch über eine gemeinsame Vorgehensweise einigen, d.h. die Zusammenarbeit als Gruppe und in der Gruppe bedarf genauer Abstimmung. Dies geschieht in den seltensten Fällen problemlos und konfliktfrei.

1.2.3 Lösungsansätze

Während die technischen Probleme zu einem Großteil durch geeignete schriftliche Anleitungen gelöst werden können, benötigen die Teilnehmer speziell bei den inhaltlichen Problemen umfangreiche Unterstützung durch einen Tutor. Dieser kann auch in Konfliktsituationen – aufgrund seiner Neutralität der Gruppe und ihren Mitgliedern gegenüber – helfend eingreifen oder aber Hinweise für eine erfolgreiche Vorgehensweise anbieten.

Hierbei ist jedoch die Verfügbarkeit des (realen) Tutors zu beachten, da dieser üblicherweise nur zu bestimmten Zeiten seine Tutorien anbieten kann und überdies meist eine Vielzahl von Gruppen bedienen muss. Die sonstigen organisatorischen Probleme lassen sich heutzutage unter Einsatz geeigneter Computersysteme bewältigen: Groupware kann sowohl bei der Terminplanung als auch bei der verteilten Bearbeitung gemeinsamer Dokumente unterstützend eingesetzt werden und dabei helfen, räumliche Distanzen zu überbrücken.

Somit entstand die Idee, eine CSCL-Umgebung, die VitaminL-IDE, zu entwickeln, die virtuelle Teams bei der gemeinsamen Programmierung in Java angemessen unterstützen soll. Es musste aber nicht nur eine integrierte Entwicklungsumgebung (Integrated Development Environment; IDE) umgesetzt werden, die die gängigen Werkzeuge der Java-Programmierung zur Verfügung stellt, und darüber hinaus die verteilte Programmierung ermöglicht, sondern es galt auch, eine tutorielle Komponente zu definieren,

die möglichst jedem Team adäquate Unterstützung anbieten soll. Dazu ist die personelle Zusammensetzung eines Teams zu berücksichtigen.

1.2.4 Die Architektur der VitaminL-IDE

Das geplante CSCL-System war von vornherein als Client-Server-System konzipiert. Die Umsetzung fand (und findet noch immer) in Java statt. Die Client-GUI ist mit den Swing-Klassen realisiert und bietet Werkzeuge zur Kommunikation, zur verteilten Handhabung von Dokumenten (Java-Quelltexte) und zum Aufruf von Java-Compiler und Java-Interpreter. Der Datenaustausch zwischen dem Server und angemeldeten Clients findet mittels Nachrichten über TCP/IP-basierte Netzwerkverbindungen statt.

1.2.5 Benutzertests und Fragebögen

Die gesamte Entwicklung des geplanten CSCL-Systems war begleitet von Evaluationen mittels Benutzertests und diversen Fragebögen. Die Benutzertests finden stets unter Einsatz der VitaminL-IDE statt: virtuelle Teams aus zwei bis vier Mitgliedern bearbeiten in einem Zeitraum von ca. zwei Stunden ausgewählte Übungsaufgaben. Ein Tutor ist dabei zwar anwesend, er greift allerdings nur bei technischen Problemen ein, so dass die eigentliche Bearbeitung der Aufgabe selbständig durch das Team erfolgt. Mit dieser Vorgehensweise sollen gezielt Problemsituationen und Lösungsstrategien der Teams aufgedeckt werden. Da außerdem sämtliche Benutzeraktionen auf dem Server protokolliert werden, können diese Daten für die Modellierung der Tutorkomponente herangezogen werden. Nicht zuletzt ist jede Sitzung auch ein Test der Software unter realen Bedingungen, dessen Ergebnisse zur Korrektur, Stabilisierung und Weiterentwicklung des CSCL-Systems verwendet werden.

Im Anschluss an eine Sitzung erhalten die Teilnehmer einen Fragebogen mit Fragen zur Sitzung und der verteilten, virtuellen Zusammenarbeit, zu aufgetretenen Problemen sowie zur Benutzungsoberfläche und der Bedienung der VitaminL-IDE. Die ausgefüllten Fragebögen enthalten Hinweise über die Akzeptanz dieser Form der Zusammenarbeit sowie über aufgetretene Probleme und wünschenswerte Funktionalitäten der IDE.

1.2.6 Der Prototyp: VitaminL-IDE V1.0

Im September 2003 wurde – nach fast 2 Jahren Recherche, Entwicklung und internen Tests – der erste Prototyp der VitaminL-IDE fertiggestellt und musste sich in einem ersten Benutzertest bewähren: zwei Studierende sollten eine kleine Schachuhr in Java entwickeln – unter Nutzung der VitaminL-IDE. Der Funktionsumfang dieser Version war noch recht begrenzt: es gab einen einfachen Chat für die Kommunikation und Basisfunktionen (wie Laden,

Speichern, Übersetzen, Anfordern und Freigeben) von einem (1!) Dokument. Es folgten drei weitere Sitzungen mit insgesamt elf Teilnehmern.

Während der Sitzungen mit dem Prototypen traten Abstürze auf, die durch Netzwerkprobleme begründet waren. Ferner wurden sowohl das GUI-Design der IDE (u.a. zu wenig Platz für das Dokument) als auch deren Funktionsumfang kritisiert. Trotz der aufgetretenen Fehler und Probleme wurde diese Form der Gruppenarbeit positiv erwähnt. Auch die räumliche Unabhängigkeit der Gruppenmitglieder sowie die Kombination aus Chat und verteilter Dokumentenbearbeitung wurde als motivationsfördernd genannt.

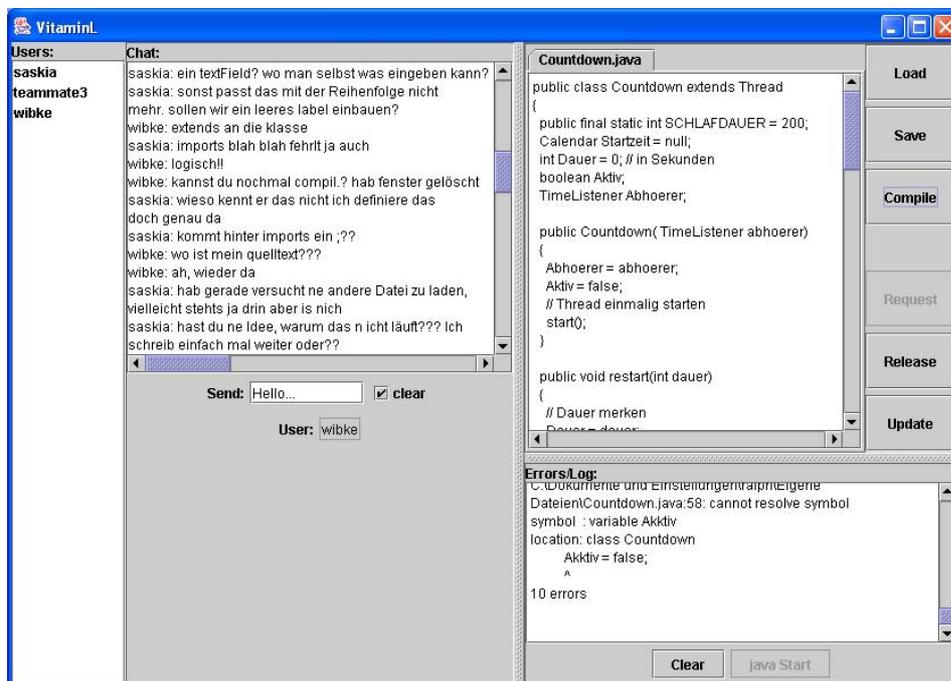


Abbildung 1: Screenshot des ersten VitaminL-Benutzertests

Insgesamt konnte mit diesem Prototypen die prinzipielle Machbarkeit des geplanten Systems überprüft werden. Obwohl Funktionalität und Design überarbeitet werden mussten, konnte die Akzeptanz dieses Ansatzes gezeigt werden.

1.2.7 Die Weiterentwicklung: VitaminL-IDE V2.0

Die Ergebnisse der ersten Benutzertests flossen in die Entwicklung der nächsten Version der IDE ein, die sich unmittelbar daran anschloss und im Juni 2004 mit fünf weiteren Benutzertests und elf Teilnehmern endete. Mit dieser Version wurde die Client-GUI überarbeitet und erweitert: ein Dokumentenmanager auf dem Server erlaubte dem Team das gleichzeitige Bearbeiten mehrerer Dokumente und der Chat wurde durch eine Dialog-Schnittstelle mit strukturierter Kommunikation ersetzt. Basierend auf den sog.

Collaborative Learning Skills (CLS) von McManus & Aiken (1995) und darauf aufbauend den Forschungen Jermann et al. (2001) werden diverse Satzanfänge angeboten, die durch den Benutzer sinngemäß vervollständigt werden müssen (vgl. Soller, 2004).

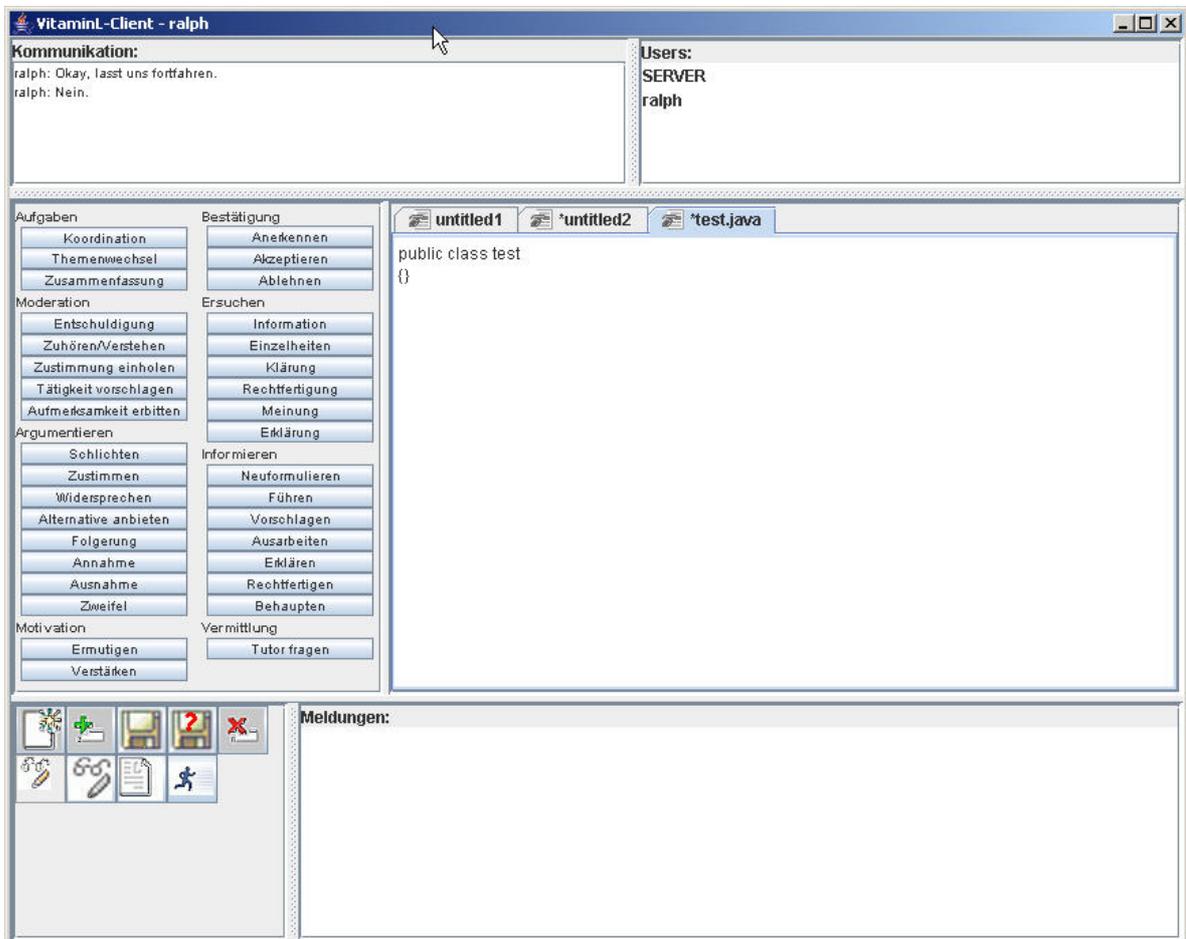


Abbildung 2: Screenshot des VitaminL-Clients V2.0

Erwartungsgemäß wurde die neue Form der Kommunikation als ungewohnt und zeitaufwendig kritisiert. Ferner wurden erstmalig Vergleiche mit der Eclipse-IDE angestellt und dabei u.a. die Leistungsfähigkeit des Editors von VitaminL bemängelt. Trotz dieser Kritikpunkte und teilweise aufgetretener Performance-Probleme wurden das *Document Sharing* und damit verbunden auch die besseren Möglichkeiten des Wissensaustauschs innerhalb der Teams positiv hervorgehoben. Das Prinzip der strukturierten Kommunikation ist also durchaus einsetzbar (vgl. auch Baker & Lund, 1997).

1.2.8 Das Re-Design: VitaminL-IDE V2.5

Basierend auf den eben genannten Kritikpunkten begann im Juli 2004 eine vollständige Überarbeitung der IDE, in deren Folge nicht nur die Client-GUI komplett neugestaltet wurde, sondern auch wesentliche Teile des Quellcodes

neu verfasst wurden. Abbildung 3 zeigt, dass im Mittelpunkt der IDE die Dokumente (Quellcode) stehen, die Kommunikation wird in einem schwebenden Fenster abgewickelt, das bei Bedarf ein- und ausgeblendet werden kann. Dadurch entstand mehr Fläche für die eigentliche Bearbeitung. Diese neue Version 2.5 wurde im Oktober 2004 fertiggestellt. Im Wintersemester 2004/2005 fanden insgesamt 25 Benutzertests mit 10 Teams und 30 Teilnehmern statt.

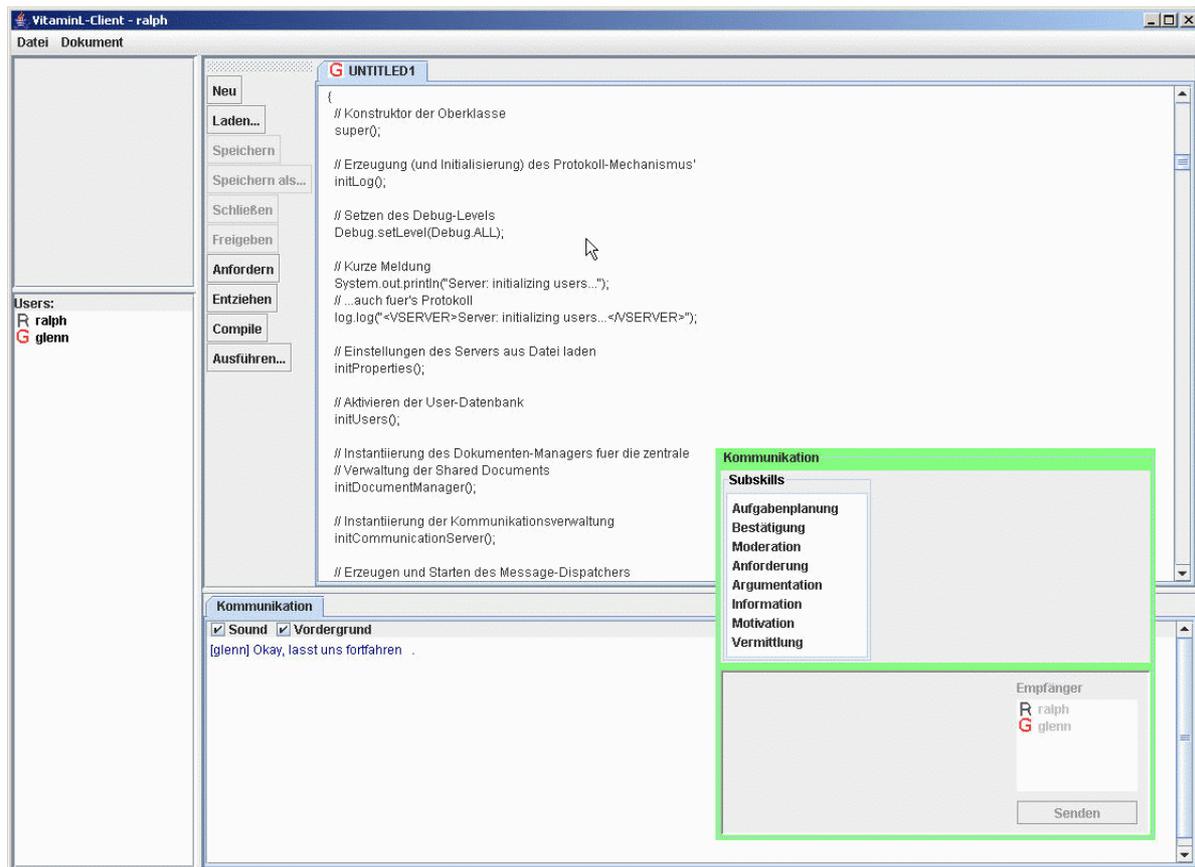


Abbildung 3: Screenshot des VitaminL-Clients V2.5

Erstmalig nahmen ganze Gruppen mehrmals an den Tests teil (mit wechselnden Aufgaben) und hatten sogar Gelegenheit, Teile ihrer Semesterprojekte mittels der VitaminL-IDE zu bearbeiten. In dieser Phase konnten nicht nur – aufgrund der umfangreichen Tests – weitere Fehler und konzeptionelle Schwächen aufgedeckt und bereinigt werden, es konnten auch viele Sitzungsdaten protokolliert werden. Diese Datenbasis bildet die Grundlage für die noch zu entwickelnde Analysekomponente des geplanten tutoriellen Systems innerhalb der VitaminL-Applikation.

1.2.9 Die aktuelle Version: VitaminL-IDE V2.99.2

Während die o.a. Versionen der VitaminL-IDE nur im Rahmen von Benutzertests in lokalen Netzen der Universität eingesetzt wurden, wurde diese Version erstmals als echte Internet-Anwendung konzipiert, bedingt durch die Tatsache, dass in der Kooperationsveranstaltung echte standortübergreifende virtuelle Teams gebildet werden sollten.

Vorbereitend mussten dazu wichtige Änderungen und Ergänzungen durchgeführt werden. Zunächst wurde das Netzwerkprotokoll (bisher: RMI; *Remote Method Invocation*) durch HTTP-basierte Verbindungen ersetzt, um möglichen Problemen durch Firewalls vorzubeugen und die Installation für den Benutzer zu vereinfachen. Ferner musste ein Konzept für die Verwaltung von Gruppen entwickelt und umgesetzt werden: in der aktuellen Version V2.5 befanden sich alle am Server angemeldeten Benutzer im selben Kontext, eine Aufteilung in verschiedene Arbeitsgruppen existierte nicht.

Zusätzlich zum regelmäßigen Einsatz im Rahmen der Kooperationsveranstaltung wurden erstmalig von den Studierenden auch Hausübungen mit der VitaminL-IDE bearbeitet. Ergänzend dazu fanden elf weitere Benutzertests mit 32 Teilnehmern statt (Stand: Juli 2005). Zusammen mit den Ergebnissen aus den zugehörigen Fragebögen wurde parallel die VitaminL-IDE weiterentwickelt und hat nun die Version 2.99.2 erreicht (Stand: Juli 2005).

Die grundsätzliche Anordnung der Oberflächenkomponenten hatte sich in Version 2.5 bewährt und wurde nicht geändert. In Abbildung 4 erkennt man aber dennoch einige grundlegende und konzeptionelle Verbesserungen der GUI. So beherrscht der Editor nun *Syntax Highlighting* und Zeilennummern für Java-Quelltexte, die aktuell vom Besitzer der Datei bearbeitete Zeile wird markiert, sodass alle Mitglieder des virtuellen Teams sofort erkennen können, wo gerade geändert wird. Desweiteren wurde ein neues Navigationselement zur Verfügung gestellt, mit dem man schnell zur bearbeiteten Zeile springen kann. Möchte der Besitzer einer Datei etwas erklären, so kann er Markierungen im Quelltext setzen, die an die anderen Teammitglieder übertragen werden, sodass sich die Kommunikation über den Quelltext weitgehend auf die wichtigen Inhalte konzentrieren kann. Ferner wurde die Ausführung von Java-Programmen durch Aufruf des Java-Interpreters in die IDE integriert, Benutzer mit Tutorrechten können die Gruppen wechseln und bei Wegfall der Netzwerkverbindung zum Server kann ein Client lokale Kopien aller geöffneten Dokumente abspeichern.

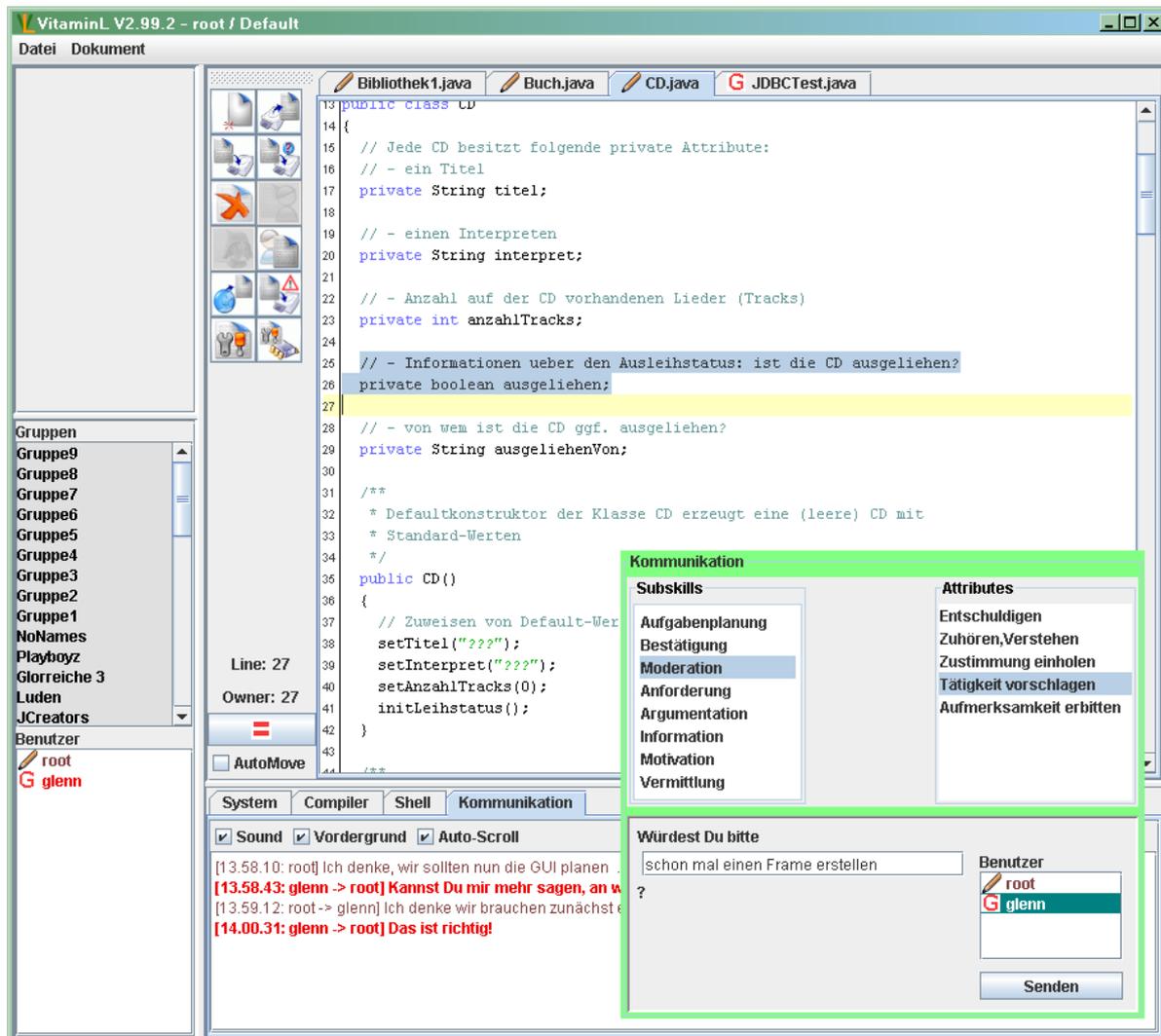


Abbildung 4: Screenshot des VitaminL-Clients V2.99.2

Grundsätzlich muss angemerkt werden, dass technische Probleme gleich welcher Art sowohl die Akzeptanz eines solchen Systems als auch die Motivation, damit zu arbeiten, stark negativ beeinflussen können. Ferner ist zu berücksichtigen, dass neuartige Bedienkonzepte nicht nur mit einem gewissen Mehraufwand bei der Einarbeitung behaftet sind, sondern dass den Benutzern der Sinn und Zweck dieser neuen Konzepte auch glaubhaft vermittelt werden kann.

Mit der Version 2.99.2 der VitaminL-IDE liegt ein technisches CSCL-System für die verteilte, synchrone Programmierung in Java vor, das dank einer ständigen, begleitenden Evaluation einen stabilen Stand erreicht hat und mittels einer noch zu entwickelnden Tutorenkomponente zu einem ITS erweitert werden wird.

1.3 Evaluation

Wie schon erwähnt wurde VitaminL in jeder Version von einer Evaluation begleitet. Neben den Fragebögen und Interviews wurden die Benutzertests nachträglich einer intellektuellen Logfile-Analyse unterzogen. Dazu wird jede Aktion, die innerhalb des VitaminL-Tests stattfindet, in einem Logfile gespeichert. Ein Logfile-Analyser, der ebenfalls im Projekt entwickelt wurde, bietet dazu die Möglichkeit, jederzeit einen Benutzertest nachträglich wieder „abzuspielen“ und so ggf. Probleme aufzudecken. Falls also während eines Benutzertests etwas übersehen wird, kann er im Prinzip jederzeit wiederholt werden.

Eine empirische Evaluation fand im Sommersemester 2005 während der schon erwähnten Kooperationsveranstaltung der Universitäten Konstanz und Hildesheim statt. Die Teilnehmer in Hildesheim gehörten dem Studiengang Internationales Informationsmanagement (IIM) an, die in Konstanz den Studiengängen Information Engineering (BA und MA), International Economic Relations (MA) und Biologie. Somit war von vornherein klar, dass die Gruppen sehr interdisziplinär ausgerichtet sein und sich die Vorkenntnisse auf sehr unterschiedlichem Niveau befinden würden.

Der didaktische Ansatz der Veranstaltung war der des Blended-Learning. Die Vorlesungsinhalte wurden per Video-Konferenz (Polycom Videostation FX) und VNC (Virtual Network Computing) nach Konstanz übertragen. Beide Verfahren sind etabliert und funktionierten problemlos. Dabei kamen den Dozenten Erfahrungen aus den Projekten MEUM und Virtueller Campus, die in den letzten Jahren in der Informationswissenschaft durchgeführt worden waren, sehr zugute. Die allgemeine Meinung, virtuelle Veranstaltungen generierten einen höheren Aufwand als traditionelle, bestätigte sich nicht.

Wichtig bei gemeinsamen Veranstaltungen verschiedener Standorte ist die effiziente Verwaltung der Teilnehmer, der Unterrichtsmaterialien und der Übungsaufgaben in einem Lernmanagement-System (LMS). Die Aufgabe übernahm das im Rahmen des K3-Projekts in Konstanz entwickelte K3-forum. Dieses System bietet darüber hinaus eine effiziente Gruppenverwaltung und Werkzeuge zur asynchronen Kommunikation wie Email oder moderierte und unmoderierte Foren. Da sich fünf der insgesamt acht Gruppen aus Teilnehmern aus Hildesheim und Konstanz zusammensetzten, war eine besondere und neue Herausforderung die Durchführung der virtuellen Programmier-Tutorien mit dem VitaminL-System. Bzgl. der Weiterentwicklung des K3 sind Module in Entwicklung,

die die Mitarbeit der Studierenden im System automatisch bewerten. Der zweite Teil dieses Beitrags geht darauf näher ein. Bei VitaminL sind o.a. Module konzipiert, die den Gruppen in Problemsituationen, die insbesondere bei Programmieranfängern zwangsläufig auftreten, automatisch geeignete Hilfe zur Verfügung stellen werden.

Wie schon die VitaminL-IDE selbst wird sich auch die Tutorkomponente – und darin enthalten die Analysekomponente – einer qualitativen Überprüfung unterziehen müssen. Dazu werden (voraussichtlich im Wintersemester 2005/2006) weitere Benutzertests im Rahmen der Java-Veranstaltungen in der Universität Hildesheim stattfinden. Die Ergebnisse dieser Tests werden – neben der Qualitätsüberprüfung – auch zur Qualitätssteigerung der Analyseverfahren verwendet werden, um wie in dem umgebenden System, der VitaminL-IDE, auch in den neuen Systemkomponenten durch ständige Tests und Verbesserungen einen hohen Entwicklungsstandard und Resultate hoher Güte zu erzielen.

1.2.4 Ausblick

„Stillstand bedeutet Rückschritt.“² Der nächste wichtige Schritt bei der Weiterentwicklung von VitaminL besteht in der Umsetzung der Tutorenkomponente. Eine der Herausforderung wird dabei sicher in der verlässlichen und zeitnahen Erkennung von Benutzerprofilen bestehen, da die Teamsitzungen mit der VitaminL-IDE oftmals nicht länger als zwei Stunden dauern. Auf Basis der Benutzerprofile soll der virtuelle Tutor das Team gezielt unterstützen. Gezielt bedeutet hierbei, dass auf Basis der Benutzerprofile entschieden wird, ob helfend eingegriffen wird oder nicht. Zu wenig Hilfe untergräbt die Motivation des Teams, zu viel Hilfe verhindert u.U. den Lernerfolg.

Ferner ist eine Erweiterung des Funktionsumfangs geplant: neben einer Integration des Java-Debuggers zur Fehlersuche in laufenden Programmen wird derzeit – speziell für die Entwurfsphase von Projekten – ein sog. *Shared Whiteboard* konzipiert. Ergänzend zur textbasierten Kommunikation ermöglicht solch eine Komponente die verteilte Erstellung von Grafiken wie einfachen Entwurfsskizzen oder aber auch komplexeren UML-Diagrammen. Damit wird dem Wunsch zahlreicher Benutzer entsprochen.

² Erich Kästner

Eine weitere interessante Variante der VitaminL-Applikation würde ein entsprechendes Plugin für Eclipse darstellen. Auf diese Weise könnten die Vorteile beider Umgebungen miteinander kombiniert werden. Eclipse ist eine leistungsfähige Entwicklungsumgebung (nicht nur) für die Programmierung in Java, die durch ein Plugin-Konzept erweiterbar ist und sich so individuell einrichten lässt. Der Vorteil der VitaminL-IDE liegt weniger im Funktionsumfang als vielmehr in der Unterstützung virtueller Teams bei der verteilten, synchronen Java-Programmierung durch einen virtuellen Tutor, der gezielten Support unter Berücksichtigung der personellen Zusammensetzung des Teams anbieten soll.

2 Kollaborative Leistungsevaluation beim Einsatz von Wissensmanagementsystemen in der Ausbildung

2.1 Motivation

Neue Formen der Aus- und Weiterbildung prägen derzeit die öffentliche und universitäre Diskussion. Im Kontext von Lehren und Lernen mit neuen Medien stellt die Förderung des kollaborativen, selbstgesteuerten Lernens ein zentrales Forschungsgebiet dar [Arnold 2003, S. 16; Mandl 2000, S. 3; Semar 2004a, S. 255]. Dabei hat sich gezeigt, dass eine aktive Förderung und Unterstützung der Systemteilnehmer notwendig ist, in der Regel geschieht dies, indem die einzelnen Akteure dazu aufgefordert werden, meist persönlich durch den Dozenten oder durch ein anderes Gruppenmitglied, bestimmte Tätigkeiten auszuführen. Zwar kann dadurch ein bestimmtes Verhalten in Lernumgebungen „erzwungen“ werden, aber besser und erfolgreicher ist es, wenn die Akteure dazu motiviert werden können. Diese Motivation kann mittels entsprechender Anreizsysteme geschehen. Die Problematik liegt jedoch in der Ausgestaltung solcher Anreizsysteme [Schanz 1991, S. 23]. Wir wollen ein „Anreizsystem“ aufzeigen, das aufgrund bestimmter Kennzahlen und deren Visualisierung den einzelnen Akteur so beeinflusst, dass seine intrinsische Motivation gefördert wird und daraus eine eigenständige, aktive Teilnahme am kollaborativen Wissenserarbeitungsprozess resultiert.

Ein dem kollaborativen Ansatz angemessenes Anreizsystem mit starker reputativer Ausrichtung soll den gesamten Wissenserarbeitungsprozess unterstützen. Den Akteuren wird eine Aufgabe gestellt, die sie eigenständig als Gruppe lösen sollen. Individuelle Anreize animieren die Akteure bestimmte Leistungen zu erbringen, die dann mit Hilfe des

Bewertungssystem bewertet werden. Das Ergebnis wirkt auf die Motivation der Akteure und veranlasst sie weiterhin mitzuarbeiten, bis die gestellte Aufgabe (zufriedenstellend) gelöst wurde.

Dieses Anreizsystem³ ist seit dem Sommersemester 2005 Teil des Software-Systems K3, das in der studentischen Hochschulausbildung kollaboratives und verteiltes Erarbeiten von konzeptuellem Wissen über heterogene Ressourcen und moderierte elektronische Kommunikationsforen ermöglicht. Das Anreizsystem wird durch eine in das System integrierte Crediting-/Rating-Komponente realisiert. Durch diese Komponente werden jede noch so kleine Beiträge der Studierenden, z.B. ein Kommentar zu einem Eintrag in einem Kommunikationsforum oder das kommentierte Einbringen einer externen Verknüpfung, erfasst, die dann zum Einen dem Beitragenden angerechnet werden, entweder als individuelle Leistung oder als Teil der Gruppenleistung, und zum Anderen werden durch diese Aktivitäten bestimmte Kennzahlen generiert und visualisiert. Auf diese Weise erhalten die Lernenden ein permanentes Feedback, das ihnen eine ständige Kontrolle über ihre eigenen Leistungen ermöglicht. Durch den Vergleich seiner Leistungen mit denen seiner Kommilitonen erhält jeder Akteur zur jeder Zeit einen Überblick über seinen Stand in der Community. Somit wird eine dynamische und individuelle, aber auch das Gruppenverhalten berücksichtigende Evaluierung des Lernerfolgs über eine aktive Teilnahme ermöglicht [Semar 2004b].

2.2 Entwicklung eines Kennzahlensystems zur Evaluation kollaborativer Gruppenarbeit

Kollaboration ist ein komplexer Vorgang, entsprechend schwierig ist es festzustellen, wann eine Gruppe kollaborativ bzw. nur kooperativ arbeitet, oder wie effektiv diese Arbeit ist, oder was kollaboratives Arbeiten unterstützt bzw. es verhindert. Verschiedene Autoren haben unterschiedliche Merkmale erarbeitet, mit deren Hilfe man kollaborative Gruppenarbeit feststellen kann.

Das erste Merkmal ist die „Wechselbeziehung“ (interdependence) in einer Gruppe [Johnson 1998]. Eine Wechselwirkung wird erreicht, wenn sich jedes

³ K3 wird zur Zeit an der Universität Konstanz am Lehrstuhl Informationswissenschaft (Prof. Kuhlen) entwickelt. Es handelt sich dabei um ein vom BMBF (DLR PT-NMB+F) im Rahmen des Programms „Innovation und Arbeitsplätze in der Informationsgesellschaft des 21. Jahrhunderts“ in Bezug auf die Fachinformation gefördertes Projekt (Projektnummer: 08C5896). Unter <http://www.k3forum.net> sind weitere Informationen zu erhalten.

Gruppenmitglied aktiv an der Diskussion beteiligt. Auf einer etwas oberflächlichen Ebene reicht dafür die einfache Teilnahme jedes Mitglieds in der Gruppe aus. Eine annähernd gleich verteilte Teilnahme der Mitglieder ist sicherlich eine notwendige aber nicht ausreichende Voraussetzung für eine erfolgreiche Wechselbeziehung. Um festzustellen, ob eine gleichverteilte Wechselbeziehung in der Gruppe vorherrscht, muss überprüft werden, ob die Mitglieder auch wirklich interagieren. Die Wechselbeziehung lässt sich somit an zwei notwendigen Eigenschaften identifizieren. Das ist zum Einen die Teilnahme (*participation*) an der Gruppe und zum Anderen die Interaktion (*interaction*), eine aktive Teilnahme, in der Gruppe. Die Teilnahme ist notwendig, da ohne sie keine wirkliche Kollaboration in der Gruppe entsteht. Kollaboration kann nur erreicht werden, wenn die Gruppenmitglieder annähernd gleich an der Gruppenarbeit teilnehmen. Bei der Interaktion handelt es sich um mehr als eine einfache Teilnahme, es geht hier um eine Aktion, im Sinne von mit jemandem agieren bzw. auf jemanden reagieren [Ingram 2003; S. 225]. Findet keine aktive, reagierende oder gar den Diskurs verändernde Interaktion statt, kann nicht von Kollaboration sondern maximal von Kooperation gesprochen werden.

Das zweite Merkmal setzt voraus, dass die Gruppe gemeinsam an einem Ziel arbeitet. Kollaboratives Arbeiten ist dabei mehr als das einfache Austauschen von Meinungen und Ideen. Das daraus resultierende Ergebnis ist eine Synthese aus der gemeinsamen Diskussion bzw. aus allen individuellen Beiträgen. Die Kombination der individuellen Beiträge führt zu einem Produkt, das mehr ist als die Summe der einzelnen Beiträge [Kaye 1992, Henri 1992]. Dieses Produkt bildet eine Synthese des gemeinsam erarbeiteten Wissens und unterscheidet sich von jenem, das jedes Gruppenmitglied alleine produziert hätte.

Das dritte Merkmal ist die Unabhängigkeit der Gruppe vom Dozenten, unabhängig im Sinne von autonom. Auf der einen Seite muss die Gruppe ohne dass sie die Hilfe des Dozenten in Anspruch genommen hat, eigenständig (*independent*) zur Synthese gekommen sein [Laffey 1998]. Auf der anderen Seite ist es notwendig, dass der Dozent nicht korrigierend in die Gruppenarbeit eingreifen musste. Die Gruppe muss also fähig sein, miteinander zu kollaborieren und alternative Quellen zu suchen um selbständig zu einer Synthese zu kommen. Diese letztendlich vier Eigenschaften (Teilnahme, Interaktion, Synthese und Unabhängigkeit) können im K3-System durch bestimmte Kennzahlen ermittelt werden.

Der Begriff *Kennzahl* hat eine vielseitige Entwicklung durchlaufen, an deren Ende ein allgemein akzeptierter Kennzahlenbegriff steht [Reichmann 2001, S. 19]. Seit Mitte der 70er Jahre besteht in der Literatur eine weitgehende Einigkeit darüber, dass Kennzahlen komplexe, quantitative Sachverhalte, die Teilbereiche eines Untersuchungsbereichs betreffen, in konzentrierter Form abbilden [Schwickert 2000, S. 3]. Charakterisiert werden Kennzahlen durch ihren Informationscharakter, die Quantifizierbarkeit und ihre spezifische Form. Der Informationscharakter zeigt sich darin, dass Kennzahlen wichtige Sachverhalte und deren Zusammenhänge aufzeigen. Die Quantifizierbarkeit ist eine Eigenschaft von Variablen, die den Sachverhalt und die Zusammenhänge, messtheoretisch gesehen, auf ein metrisches Skalenniveau bringen und damit relativ präzise Aussagen ermöglichen. Die spezifische Form ermöglicht komplizierte Strukturen und Prozesse auf relativ einfache Weise darzustellen, zum Zwecke eines schnellen und umfassenden Überblicks [Reichmann 2001, S.20].

Die Aussagekraft einzelner Kennzahlen ist, ohne den gedanklichen Hintergrund zu kennen, begrenzt [BIFOA 1980, S. 20], darum besteht die Gefahr einer inadäquaten Interpretation der Einzelkennzahlen. Aus diesem Grund empfiehlt sich dort, wo ergänzend qualitative Informationen notwendig sind, eine kombinierte Anwendung von quantitativen und qualitativen Informationen [Staehe 1969]. Der mangelnden Aussagekraft einzelner Kennzahlen wird durch die Kombination eines ausgewählten Satzes von Kennzahlen entgegengewirkt. Dabei ist es sinnvoll, mehrere, sachlich zusammenhängende Kennzahlen zu einem Kennzahlensystem zu verbinden, das die Beziehungen und gegenseitigen Wirkungen der einzelnen Kennzahlen darlegt. Die Beziehungen zwischen den Kennzahlen eines Systems basieren entweder auf einem mathematischen oder einem sachlogischen Zusammenhang [Meyer 1994]. Allgemein wird unter einem Kennzahlensystem „eine Zusammenstellung von quantitativen Variablen verstanden, wobei die einzelnen Kennzahlen in einer sachlich sinnvollen Beziehung zueinander stehen, einander ergänzen oder erklären und insgesamt auf ein gemeinsames übergeordnetes Ziel ausgerichtet sind“ [Reichmann 2001, S. 23].

2.3 K3-Kennzahlensystem zur Messung von kollaborativer Gruppenarbeit

Kuhlen beschreibt 1998 erstmalig unterschiedliche Aktivitätsgrade, die er zur Messung der Informations- und Kommunikationsbereitschaft in elektronischen Kommunikationsforen benutzt [Kuhlen 1998, S. 50 f].

Aufbauend auf dieser Grundlage wollen wir hier weitere Kenngrößen erarbeiten, die in K3 als Messgrößen eingesetzt werden. Dabei gilt es zu beachten, dass zu viele Kennzahlen, vor allem in unbearbeiteter Form, zu einer Überversorgung des Informationsempfängers führen. Zur bedarfsgerechten Informationsversorgung werden die Kennzahlen daher in einem Kennzahlensystem verdichtet. In einem weiteren Schritt werden diese Maßzahlen mit Hilfe angemessener Visualisierungsformen dargestellt.

Zum Aufbau eines (Anreiz-, Motivations-)Kennzahlensystems in K3 ist ein hierarchisches System nicht geeignet, da die Messgrößen aus dem K3-System nur teilweise in einem mathematischen Zusammenhang stehen. Sinnvoller erscheint eine Gliederung der beobachtbaren Messgrößen nach sachlichen Kriterien [Hummel 2003, S. 555]. In Anlehnung an das von Grob et al. [Grob 2004, S. 50 f.] vorgestellte grundlegende Kennzahlensystem für Learning Management Systeme (LMS), können in K3 die Messgrößen *Bestands-*, *Verhältnis-* und *Zeitraumgrößen* unterschieden werden, die auf vier verschiedenen Ebenen, der *System-*, *Kurs-*, *Gruppen-* und *Individualebene* erfasst werden.

Bestandsgrößen werden aus den beobachtbaren Messgrößen, etwa die Anzahl der Teilnehmer und Beiträge, als absolute Zahlen in Form von Einzel- oder Summenkennzahlen gebildet. Durch die Kombination absoluter Zahlen entstehen *Verhältnisgrößen*, sie können in Form von prozentualen Anteilswerten oder Indexzahlen abgebildet werden [Schwickert 2000, S. 8]. *Zeitraumgrößen* werden aus der Beobachtung des Nutzungsverhaltens über einen längeren Zeitraum abgeleitet. Mit Hilfe der Zeitreihenanalyse können Veränderungen einer Kennzahl über einen Zeitraum untersucht werden.

Das didaktische Konzept in K3 basiert auf einer kollaborativen Gruppenarbeit. Eine Gruppe bekommt vom Dozenten in einem Kurs (*Kursebene*) einen Arbeitsauftrag gestellt, den sie dann durch eigene Wissensarbeit erfüllen muss (*Gruppenebene*). Die einzelnen Gruppenmitglieder (*Individualebene*) müssen sich zunächst unterschiedliche *Rollen* (Präsentator, Rechercheur, Moderator und Summarizer) zuweisen, die sie dann während eines Arbeitsauftrages inne haben. Der Aushandlungsvorgang wird der Gruppe selbst überlassen. Die Diskussion bzw. die einzelnen Beiträge zu diesem Aushandlungsvorgang müssen von den Studenten als Beitrag vom Typ „Organisationelles“ gekennzeichnet werden. Dies geschieht durch das Auswählen eines entsprechenden Auszeichnungsfeldes im K3-System. Zusätzlich zu der jeweiligen Rolle soll jedes Gruppenmitglied aktiv an der Diskussion zum Arbeitsauftrag

mitwirken, um so die kollaborative Wissensarbeit zu gewährleisten, auch hierbei muss jeder einzelne Beitrag entsprechend seinem Typ gekennzeichnet werden. In K3 stehen folgende Beitragstypen zur Auswahl: Resultat (Präsentation, Zusammenfassung), Ergänzung, Organisationelles, Neues Thema, These, Frage, Kritik. Jeder einzelne Beitrag erhält somit ein spezielles Merkmal, je nachdem, um welchen Typ von Beitrag es sich handelt und wer ihn verfasst hat.

In diesem Paper beschränken wir uns auf die Beschreibung der Kennzahlen auf der Gruppen- und Individualebene, die zur Feststellung des „Grades der Kollaboration einer Gruppe“ dienen. Da der Übergang von kooperativem zu kollaborativem Arbeiten fließend ist, ist es nicht einfach mit Hilfe einer einzigen Kenngröße auf die Arbeitsweise einer Gruppe schließen zu können. Der hier entwickelte Gruppenkollaborationsgrad, darf aus diesem Grund nicht isoliert, sondern immer im Zusammenhang mit anderen Kenngrößen betrachtet werden.

Auf der Gruppen- und Individualebene handelt sich neben einigen organisatorischen hauptsächlich um didaktische Kennzahlen. Gerade die didaktischen Größen können als Anreize zur Motivationssteigerung angesehen werden, aus diesem Grund ist es hier besonders wichtig diese Kennzahlen ins Verhältnis zu den anderen Gruppen und deren Mitglieder zu setzen, da sich so jede Gruppe ein Bild über ihre Position im Verhältnis zu den anderen Gruppen machen kann. Aber auch die zeitliche Veränderung dieser Größen ist für die Gruppe und deren Akteure wichtig, dadurch kann ihnen ihre eigene Entwicklung aufgezeigt werden. Gerade die Messgrößen auf der Gruppenebene bilden die eigentlich interessantesten Kenngrößen, denn auf dieser Ebene beschäftigen wir uns mit dem eigentlichen kollaborativen Wissensmanagement. Die Kollaboration in der Gruppe lässt sich, wie oben gezeigt, an den vier Merkmalen Teilnahme, Interaktion, Synthese und Unabhängigkeit feststellen. Wir ermitteln für eine Gruppe jedes dieser Merkmale und setzen diese Einzelgrößen zum „*Grad der Kollaboration einer Gruppe (Gruppenkollaborationsgrad)*“ zusammen.

Die Synthese lässt sich nur durch eine kognitive Leistung überprüfen. In K3 wird dieses Problem durch die Mitglieder der Gruppe selbst gelöst. Bevor die Zusammenfassung der kollaborativen Wissensarbeit als Synthese veröffentlicht wird, müssen alle Gruppenmitglieder mit Hilfe eines in K3 realisierten Abstimmungstools ihr Einverständnis geben bzw. eine Bewertung der Synthese durchführen. Jedes Gruppenmitglied muss seine Wissensarbeit in der Synthese wiederfinden. Fehlen Teile seiner Arbeit oder wurden sie

falsch dargestellt, muss die Zusammenfassung entweder erneut bearbeitet werden, oder sie bekommt eine entsprechend schlechte Bewertung. Sind alle Gruppenmitglieder mit der Zusammenfassung einverstanden und findet jedes Gruppenmitglied sich mit seiner Wissensarbeit in der Synthese wieder, so ist der „*Grad der Synthese einer Gruppe (Synthesegrad SG_g)*“ gleich 1. Strebt der Synthesegrad gegen 0, so kann nicht mehr von kollaborativer Gruppenarbeit gesprochen werden.

Das Merkmal Unabhängigkeit beschreibt das Vermögen (Fähigkeit) einer Gruppe autonom bzw. ohne die Hilfe des Dozenten zum Ergebnis gekommen zu sein. Um dies festzustellen wird untersucht, wie stark der Dozent auf die Teilnahme, auf die Interaktion und auf den Inhalt der Diskussion Einfluss genommen hat. Eine Diskussion, in der nur wenig oder überhaupt keine Diskussionsthreads ohne die korrigierende Einflussnahme des Dozenten vorhanden sind, ist nicht unabhängig und kann darum nicht als kollaborativ angesehen werden. Die Unabhängigkeit einer Gruppe lässt sich ermitteln, indem gemessen wird, wie oft ein Dozent korrigierend in die Gruppenarbeit eingreifen musste, bzw. die Gruppe den Dozenten um Hilfe gebeten hatte. Damit das System die korrigierenden von den lobenden Dozentenbeiträgen unterscheiden kann, muss der Dozent seine Kommentare mit dem Typus „Kritik“ kennzeichnen. Vom System wird die Anzahl der korrigierenden Dozentenbeiträge festgestellt sowie die Anzahl aller Beiträge in der Gruppe (Studenten- plus korrigierende Dozentenbeiträge). Der „*Grad der Unabhängigkeit einer Gruppe (Unabhängigkeitsgrad UG_g)*“ wird definiert als $1 - (\text{Anzahl der korrigierenden Dozentenbeiträge} / \text{Anzahl aller Beiträge in der Gruppe})$. Geht dieser Wert gegen 0, so herrscht in der Gruppe keine Unabhängigkeit; geht er gegen 1, so ist eine ausgeprägte Unabhängigkeit vorhanden.

Eine funktionierende Interaktion erfordert zunächst einen Beitrag und einen nachfolgenden, sich auf den Beitrag beziehenden Kommentar. Isoliert stehende Beiträge sind unabhängige Beiträge, die sich nicht auf einen früheren Beitrag beziehen. Sie gehören nicht zu einem Diskussionsstrang, sie antworten nicht auf einen Kommentar noch haben sie einen Kommentar hervorgerufen. Isolierte Beiträge in K3 sind Kommentare, die noch kein Feedback bekommen haben, oder eine Frage ohne eine Antwort oder ein Thread mit nur einem Beitrag. Kollaborative Gruppenarbeit ist mehr als der Austausch von Wissen in Form von unabhängigen Beiträgen [Ingram 2003, S. 228]. Eine hohe Anzahl an isolierten Beiträgen zeichnet vielleicht kooperative Gruppenarbeit aus, jedoch nicht kollaborative. Isolierte Kommentare stellen keine Interaktion in unserem Sinne dar. Interaktion benötigt zumindest einen

Kommentar und eine Reaktion auf diesen. Eine Reaktion muss sich inhaltlich also auf einen früheren Beitrag beziehen, tut sie das nicht, ist sie keine Reaktion sondern ein isolierter Beitrag. Der Diskussionsstrang A-B-C-A beschreibt, dass drei unterschiedliche Mitglieder nacheinander einen Kommentar abgegeben haben, B und C sind eine Reaktion auf A, der letzte Kommentar wurde wieder von A abgegeben und stellt auch eine Reaktion dar. Der „*Grad der Interaktion eine Gruppe (Interaktionsgrad)*“ wird definiert als $1 - (\text{Anzahl der isolierten Beiträge} / \text{Anzahl studentischer Beiträge})$. Geht dieser Wert gegen 1, so herrscht in der Gruppe keine Interaktion, geht er gegen 0, so ist eine ausgeprägte Interaktion vorhanden.

Eine aktive Teilnahme an der Gruppenarbeit ist die notwendige Voraussetzung zur Interaktion. Ohne eine aktive Teilnahme der Gruppenmitglieder kann weder von einer Kooperation noch von einer Kollaboration gesprochen werden. Aus diesem Grund wird für jedes einzelne Mitglied ermittelt, wie hoch sein Teilnahmegrad in der Gruppe ist. Das Merkmal Teilnahme (Participation) bestimmen wir auf der Individualebene für jedes einzelne Gruppenmitglied, in dem die „*Anzahl der Beiträge des Mitglieds m* “ errechnet werden. Der Quotient aus der „*Anzahl der Beiträge des Mitglieds m* “ und der „*Anzahl studentische Beiträge*“, die in dieser Gruppe gemacht wurden, ergibt dann den „*Grad der Teilnahme des Mitglieds m in der Gruppe g (Teilnahmegrad $T_m G_g$)*“. Strebt der Wert gegen 0, so hat dieses Mitglied nichts zur Gruppenarbeit beigetragen, strebt der Wert gegen 1, so stammen alle Beiträge von diesem Mitglied und die anderen Gruppenmitglieder haben keine Beiträge geleistet. Auf der Gruppenebene vergleichen wir dieses Maß und führen die Mitarbeitsgrade jedes Gruppenmitglieds zur einer Maßzahl zusammen, die uns den „*Grad der Teilnahme der Gruppe (Gruppenteilnahmegrad TG_g)*“ angibt. In einer kollaborativ arbeitenden Gruppe müsste jedes Gruppenmitglied gleich viel teilgenommen (Beiträge geleistet) haben. In einer Gruppe von vier Mitgliedern wäre der Teilnahmegrad $T_m G_g$ gleich 0,25 ($\sigma 4$; Standardwert für eine Vierergruppe) für jedes Mitglied. Hat ein Mitglied den Grad 0, so hat dieses nicht in der Gruppe mitgearbeitet. Angenommen, die restlichen drei Mitglieder haben wieder gleich viel mitgearbeitet, so erhalten sie jeweils einen „*Teilnahmegrad*“ von 0,33 ($\sigma 3$; Standardwert einer Dreiergruppe). Die Abweichung vom Standardwert 0,25 gibt somit Auskunft über den unterschiedlichen Teilnahmegrad. Um eine einzige Kennzahl für die ganze Gruppe zu erhalten summieren wir alle Abweichungen auf.

$$TG_g := \sum_{m=1}^I |T_m G_g - \sigma_I|$$

$T_m G_g$ = Anzahl der Beiträge des Mitglieds m / Anzahl studentische Beiträge in der Gruppe g

I = Anzahl der Gruppenmitglieder der Gruppe g

σ_I = Standardwert einer Gruppe mit I Mitgliedern

Der Wert dieser Summe wird 0, wenn die Gruppenmitglieder gleichverteilt teilgenommen haben, je stärker der Wert abweicht, um so geringer war die Gleichverteilung der Teilnehmer in der Gruppe. Diese Art der Berechnung hat allerdings einen Nachteil. Der Standardwert ist abhängig von der Gruppengröße, daraus resultierend erhalten wir einen Grenzwert für TG_g vom Wert 2, und nicht, wie wir haben möchten, vom Wert 1. Wir müssen deshalb auf ein anderes mathematisches Verfahren zurückgreifen. Die mathematische Funktion der Entropie behebt dieses Problem, sie überführt eine Messreihe in die Werte von 0 bis $\ln(I)$. Wenn wir diese Entropie noch zusätzlich durch $\ln(I)$ „normalisieren“, erhalten wir einen Wertebereich von 0 bis 1 und wir können zusätzlich Gruppen unterschiedlicher Größe miteinander vergleichen.

$$hn(g) := - \sum_{i=1}^I T_m G_g \ln(T_m G_g) / \ln(I) \quad \text{mit } 0 \leq hn(g) \leq 1$$

$hn(g)$ = normalisierte Entropie der Gruppe g

Wir definieren somit den „Grad der Teilnahme der Gruppe (Gruppenteilnahmegrad TG_g)“ als die normalisierte Entropie $hn(g)$ der Gruppe g . Strebt dieser Wert gegen 1, haben alle Gruppenteilnehmer gleichverteilt teilgenommen, strebt der Wert gegen 0, liegt eine Ungleichverteilung der Teilnahme vor.

Der „Grad der Kollaboration einer Gruppe (Gruppenkollaborationsgrad GKG_g)“ ergibt sich aus dem Zusammenfügen der einzelnen Grade „Synthesegrad SG_g “, „Unabhängigkeitsgrad UG_g “, „Interaktionsgrad IG_g “ und dem „Teilnahmegrad TG_g “ zu einem Vektor. Wir erhalten somit ein Quadrupel, das den Grad der Kollaboration einer Gruppe beschreibt.

| Gruppenkollaborationsgrad GKG_g | Ideal | Wertebereich |
|-----------------------------------|--|--------------|
| Synthesegrad SG_g | $\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$ | 0 bis 1 |
| Unabhängigkeitsgrad UG_g | | 0 bis 1 |
| Interaktionsgrad IG_g | | 0 bis 1 |
| Teilnahmegrad TG_g | | 0 bis 1 |

Tab. 1: Abbildung des Quadrupel „Gruppenkollaborationsgrad“

Um eine bessere Darstellung der Werte zu erhalten, kann der Vektor bzw. die vier Grade auch visualisiert werden. In diesem Falle bietet sich eine Visualisierung mit Hilfe eines Spinnennetzdiagramms (Netzdiagramms) an, mit diesem Diagramm lassen sich quantitative Bewertungen und Einschätzungen auf einfache Weise darstellen. Ein Netzdiagramm ist eine Art Liniendiagramm, bei dem jede Kategorie ihre eigene Größenachse besitzt und jede Datenreihe wird über Linien verbunden. Aussagekräftig ist bei diesem Diagramm das eingeschlossene Gebiet im Vergleich zu anderen Datenreihen. Da die Grade unabhängig voneinander sind, ist die Reihenfolge der Achsen bzw. deren Benennung frei wählbar. Aus Gründen der Übersichtlichkeit sollte jedoch eine einmal eingeführte Reihenfolge stets beibehalten bleiben.

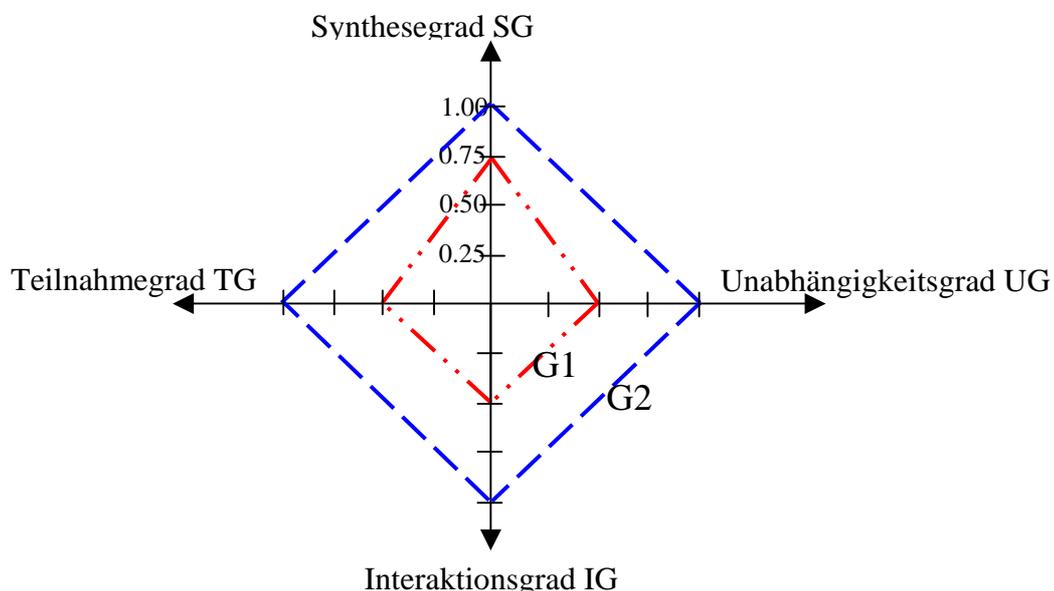


Abbildung 5: Visualisierung des Gruppenkollaborationsgrades einer Gruppe. Vergleich von 2 Gruppen: G1 (0,75/0,5/0,5/0,5) und G2 (1/1/1/1)

Beim Vergleich der Gruppen untereinander lässt sich der „Gruppenkollaborationsgrad GKG_g “ wohl am sinnvollsten ins Verhältnis zueinander setzen bzw. miteinander vergleichen. Dadurch erhalten die Gruppenmitglieder und der Dozent einen Überblick über den unterschiedlichen Kollaborationsgrad der einzelnen Gruppen. In einer Zeitreihenanalyse kann aufgezeigt werden, wie sich der Gruppenkollaborationsgrad in einer Gruppe und im Verhältnis zu den anderen Gruppen eines Kurses während der Dauer eines Arbeitsauftrages verändert hat.

2.4 Zusammenfassung

Das hier skizzierte Kennzahlensystem liefert zunächst einen Überblick über einige zentrale Messgrößen, die zur didaktischen, organisatorischen, technischen und motivationalen Steuerung in K3 eingesetzt werden. Bereits bei der Entwicklung der K3-Software wurde darauf geachtet, dass noch weitere Kennzahlen außer den hier vorgestellten „gemessen“ werden können. Es ist zu überlegen in wie weit noch Messgrößen auf der Individualebene entwickelt werden müssen, um noch mehr über das Verhalten der einzelnen Teilnehmer zu erfahren. Denkbar wären Größen wie z.B.: Wer hat in welcher Zeit auf einen Beitrag reagiert (schnellster, langsamster)? Oder das Bestimmen der Anzahl der Reaktionen auf einen Beitrag. Oder die Anzahl der durchschnittlichen Häufigkeit der Reaktionen auf einen eingegebenen Beitrag. Dies könnte ein Indiz dafür sein, dass es sich bei diesem Beitrag um ein entscheidendes Dokument handelt. Ein noch zu lösendes Problem stellt die Visualisierung der vom System erhobenen Kennzahlen dar. Welche Visualisierungsformen sind sinnvoll, überwiegend grafische Darstellungen oder doch eher eine Darstellung der Kennzahlen in tabellarischer Form? Darüber hinaus stellt sich die Frage, wann soll wer welche Kennzahlen präsentiert bekommen? Mit dem bisherigen System wurde ein Grundstock geschaffen, der sukzessive ausgebaut werden muss. Eine erste Evaluation des K3-Konzepts hat gezeigt, dass die permanente Ermittlung und Veröffentlichung der Messgrößen durchaus eine positive Wirkung auf die Mitarbeit und die Motivation der Akteure hat.

3 Fazit

Die Kombination aus K3 für die asynchrone Kommunikation, die Verwaltung der Lehrmaterialien und Bewertung der Gruppenarbeit und VitaminL als synchrones Programmierwerkzeug für virtuelle Teams hat sich auf Anhieb als sehr effizient herausgestellt, so dass eine weitere Zusammenarbeit der Universitäten bei gleichzeitiger Weiterentwicklung der Systeme sehr zu begrüßen ist. Insbesondere die Flexibilität des K3, das hier erstmals im Rahmen einer Programmier-Veranstaltung eingesetzt wurde, ist positiv hervorzuheben. Aber auch das VitaminL-System wurde nach anfänglichen technischen Problemen von den Benutzern akzeptiert und produktiv eingesetzt, was sicher als erster Erfolg zu werten ist.

4 Literaturverzeichnis

- Arnold 2003 Arnold, Patricia: Kooperatives Lernen im Internet. Münster: Waxmann, 2003
- Baker, Michael J. / Lund, Kristine (1997): Promoting reflective interactions in a computer-supported collaborative learning environment. In: Journal of Computer Assisted Learning (1997), Nr. 13, S. 175-193
- BIFOA 1880 Betriebswirtschaftliches Institut fuer Organisation und Automation: Kennzahlenbuch der Materialwirtschaft. Arbeitspapier 80-3. Köln: Universität Köln/BIFOA, 1980
- Grob 2004 Grob, Heinz Lothar; Bensberg, Frank; Dewanto, Lofi; Düppe, Ingo. Controlling von Learning Management-Systemen – ein kennzahorientierter Ansatz. In: Carstensen, Doris; Barrios, Beate (Hg.): Campus 2004. Kommen die digitalen Medien an der Hochschule in die Jahre? Münster: Waxmann, 2004, S. 46 - 56
- Henri 1992 Henri, France.: Computer conferencing and contend analysis. In: Kaye, Anthony R. (Hg.): Collaborative Learning Through Computer Conferencing. The Najaden Papers; [proceedings of the NATO Advanced Workshop on Collaborative Learning and Computer Conferencing, held in Copenhagen, Denmark, July 29 - August 3, 1991]. Berlin: Springer, 1992, S. 117 – 136
- Hummel 2003 Hummel, Thomas: Quellen und Elemente von Informationssystemen des Controlling. In: Steinle, Claus; Bruch, Heike (Hg.): Controlling - Kompendium für Ausbildung und Praxis. Stuttgart: Schäffer Poeschel, 2003
- Ingram 2003 Ingram, Albert L.; Hathorn, Lesley G.: Methodes for Analysing Collaboration in Online Communications. In: Roberts, Tim S. (Hg.): Online Collaborative Learning: Theory ond Praxice. Information Science Publishing: Hershey, 2003, S. 215 - 241
- Jermann, Patrick / Soller, Amy / Muehlenbrock, Martin (1999): From Mirroring to Guiding: A Review of State of the Art Technology for Supporting Collaborative Learning. URL <http://www.mmi.unimaas.nl/euro-cscl/Papers/197.pdf>. - Zugriffsdatum: 19.08.2003
- Johnson 1998 Johnson, David W.; Johnson, Roger T; Smith, Karl A: Cooperative learning returns to college. In: Change, 30 (4), 26-35
- Kaye 1992 Kaye, Anthony R.: Learning together apart. In: Kaye, Anthony R. (Hg.): Collaborative Learning Through Computer Conferencing. The Najaden Papers; [proceedings of the NATO Advanced Workshop on Collaborative Learning and Computer Conferencing, held in Copenhagen, Denmark, July 29 - August 3, 1991]. Berlin: Springer, 1992, S. 1-24
- Kuhlen 1998 Kuhlen, Rainer: Mondlandung des Internet. Konstanz: UVK, 1998
- Laffey 1998 Laffey, James; Tupper, Thomas; Musser, Dale; Wedman, John: A computer-mediated support system for projekt-based learning. Educational Technology Research and Development, 46 (1), 73 -86
- Mandl 2000 Mandl, Heinz; Fischer, Frank: Lernen und Lehren mit neuen Medien. Forschungsbericht Nr. 25, Juli 2000. München: Ludwig-Maximilians-Universität, Lehrstuhl für Empirische Pädagogik und Pädagogische Psychologie. München, 2000

- McManus, Margaret M. / Aiken, Robert M. (1995): Monitoring computer-based problem solving. In: *Journal of Artificial Intelligence in Education* (1995), Nr. 6, S. 307-336.
- Meyer 1994 Meyer, Claus: *Betriebswirtschaftliche Kennzahlen und Kennzahlensysteme*. Stuttgart: Poeschel, 1994.
- Reichmann 2001 Reichmann, Thomas: *Controlling mit Kennzahlen und Managementberichten*. München: Vahlen, 2001
- Schanz 1991 Schanz, Günther: Motivationale Grundlagen der Gestaltung von Anreizsystemen. In: Schanz, Günther (Hg.): *Handbuch Anreizsysteme in Wirtschaft und Verwaltung*. Stuttgart: Poeschel, 1991, S. 3 – 30
- Schulmeister, Rolf (2001): *Virtuelle Universität – Virtuelles Lernen*. München: Oldenbourg, 2001
- Schwickert 2000 Schwickert, Axel C.; Wendt, Peter: *Controlling-Kennzahlen für Web Sites*. In: *Arbeitspapiere WI*, Nr. 8/2000, (Hg): Lehrstuhl für Allg. BWL und Wirtschaftsinformatik, Johannes Gutenberg-Universität: Mainz, 2000
- Semar 2004a Semar, Wolfgang: Entwicklung eines Anreizsystems zur Unterstützung kollaborativ verteilter Formen der Aneignung und Produktion von Wissen in der Ausbildung. In: Carstensen, Doris; Barrios, Beate (Hg.): *Campus 2004. Kommen die digitalen Medien an den Hochschulen in die Jahre? Medien in der Wissenschaft*, Band 29. Münster: Waxmann, 2004. S 255 - 264
- Semar 2004b Semar, Wolfgang: Incentive Systems in Knowledge Management to Support Cooperative Distributed Forms of Creating and Acquiring Knowledge. In: Arabnia, Hamid et al. (Hg.): *Proceedings of the International Conference on Information and Knowledge Engineering - IKE'04*. Las Vegas: CSREA Press, 2004, S. 406 – 411
- Soller, Amy (2004): Computational Modeling and Analysis of Knowledge Sharing in Collaborative Distance Learning. In: *User Modeling and User-Adapted Interaction* (2004), Nr. 4, S. 351-381. Springer Science+Business Media B.V., 2004.
- Spencer, John / Pruss, Adrian (1995): *Top-Teams – Der Königsweg zu mehr Flexibilität, Effizienz und Erfolg im Betrieb*. München: Knaur, 1995
- Staehe 1969 Staehe, Wolfgang H.: *Kennzahlen und Kennzahlensysteme als Mittel der Organisation und Führung von Unternehmen*. Wiesbaden: Gabler, 1969